複数の接続表の制約を組み込んだLR表の生成とGLR法の拡張

田中 穂積　　今井 宏樹　　白井 清昭
東京工業大学大学院 情報理工学研究科
{tanaka,imai,kshirai}@cs.titech.ac.jp

1 はじめに

GLR 法 [Tomita 86] は一般の文脈自由文法 (CFG) が扱えるように, Knuth による LR 法 [Knuth 65] を拡張したものである. LR 法と同様に GLR 法でも, 与えられた CFG から構文解析で用いる動作表 (LR 表) をあらかじめ作成しておく. 先読み読み記号と現在の解析状態を用いて LR 表を検索し, 次に行う構文解析動作を決めることができる. GLR 法は, LR 表を用いて無駄のない統語解析を行うことができるので, 経験的にもっとも効率の良い構文解析アルゴリズムであるとされている.

我々はすでに, CFG の形式では記述が煩雑になる制約（たとえば隣接する形態素間の接続可能性を記述した接続制約）を LR 表に組み込み, 形態素解析と統語解析を統合して行なうシステム MSLR を開発している[1][Tanaka 95][Li 96].

ATR では, GLR 法をベースにした HMM-LR とよぶ音声認識法を開発している [Kita 91]. これは, CFG の形式で形態素を音素の列に展開しておくことにより, LR 表の先読み記号として音素がくることを利用して, 次の予測音素の精度をあげようとする方法である. 音声中の音素は, それが位置する環境（音素環境）により性質が変わる. このような音素環境に依存した音素のことを異音とよぶが, 隣接する異音間にも接続可能性に関する制約があり, これを接続表として表すことができる. このような場合には異音の接続表と形態素の接続表を二つとも LR 表に組み込めることが望ましい. この方法は [綾部 98] に示されているが, 第2節ではその概要を説明する.

2つの接続表の制約を組み込んだ LR 表には GOTO 部が含まれていないため, 既存の GLR 法による構文解析アルゴリズムを大幅に修正しなければならない [Walters 80]. この新しい解析アルゴリズムを第3節で説明する. 次の第4節ではさらに乾ら [Inui 97] が提案した確率 GLR 法に若干の修正が必要になることを説明する. 最後の第5節では, テストセットパープレキシティ[2] (test-set perplexity: 以下 TSP と略記) を用いて, 2つの接続表の制約を組み込む効果を説明する. それによれば, 既存の音声認識システムの言語モデルとして良く使われているトライグラムより小さな TSP 値が得られる.

[1] http://tanaka-www.cs.titech.ac.jp/pub/mslr/⇒index.html に公開されている（紙面の都合により⇒ で折り返している）.

2 複数の接続表の制約を組み込んだLR表の生成

本節では, まず複数の接続制約を組み込むために必要な条件を 2.1 節で述べ, 2.2 節で LR 表生成アルゴリズムの概略を示す. 2.3 節では簡単な文法を用いて, 複数の接続制約を LR 表に組み込む.

2.1 文脈自由文法の層

与えられた CFG から生成される LR 表に複数の制約を同時に組み込むための条件として, その CFG が層を持つことが必要である.

CFG $G_1 =< N_1, T_1, S, P_1 >$ $(S \in N_1, N_1 \cap T_1 = \emptyset)$ が与えられた時, 以下の条件を満たすような CFG $G_2 =< N_2, T_2, S, P_2 >$ $(S \in N_2, N_2 \cap T_2 = \emptyset)$ が存在する場合, G_1 は G_2 を層に持つと定義する.

$$T_2 \cup N_2 \subset N_1 \tag{1}$$

$$P_2 \subset P_1 \tag{2}$$

$$\alpha \stackrel{+}{\Rightarrow} \beta \quad \text{if} \quad \forall \alpha \exists \beta, S \stackrel{*}{\Rightarrow} \alpha (\alpha \in T_2^*) \wedge \beta \in T_1^* \tag{3}$$

この関係は図1のように表すことができる. G_1 の中に G_2 が包含され, T_1, N_1, T_2, N_2 が層をなして, 任意の T_2 レベルの記号列は必ず T_1 レベルの記号列に展開される.

なお, この関係はさらに再帰的に定義可能であることに注意されたい. 上の例では簡単のため2層の状態を示したが, G_2 に対しても同様の関係を持つ G_3 を定義することで3層の CFG を定義できる. さらに同

[2] 音声認識モデルの良さを表す指標の一つ.

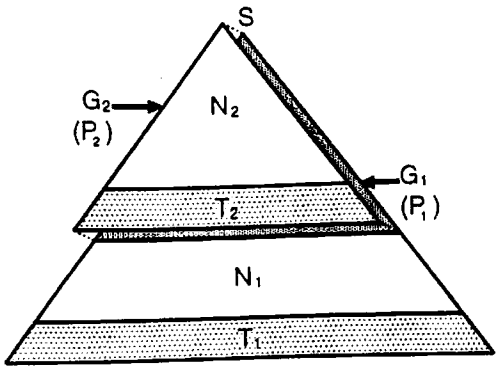

図 1: CFG の層

様の手順を繰り返すことにより，任意の数の層を持つ CFG を定義可能である．

2.2 LR 表生成アルゴリズム

以下に，2 つの接続制約を含む LR 表の生成アルゴリズム [綾部 98][Aho 86] の概略を示す．また，以下のアルゴリズムは正準 LR 表を生成する場合を示しているが，SLR 表，LALR 表の生成も同様である．

(1) G_2 に対して，既存の LR 表生成アルゴリズムを用いてクロージャ展開を行ない，GOTO グラフを生成する．

(2) 規則集合 $P_1 - P_2$ を用いて，T_2 から T_1 を導出する部分のクロージャ展開を行ない，GOTO グラフを拡張する．
その際，T_2 に属する記号を先読みに持ち，かつドットが右辺の最右端に到達しているアイテム $X \rightarrow \alpha\cdot;v_2$ $(v_2 \in T_2)$ が存在する状態に対して，先読み記号 v_2 を展開するアイテム $v_2 \rightarrow \cdot\beta;v_1$ $(v_1 \in T_1)$ を追加する．

(3) (2) で作成した GOTO グラフから以下のようにして shift 動作，reduce 動作をそれぞれ定義する．以下では，I_j, I_k は状態 j, k のアイテム集合を表す．
- GOTO(I_j,X) $= I_k$ の時，状態 j，先読み記号 X の欄に，shift k を書き込む．ただし，$X \in T_1 \cup N_1$．
- $Y \rightarrow \alpha\cdot\ ;X$ $(X \in T_1 \cup N_1)$ をアイテムに持つ I_j に対して，状態 j，先読み記号 X の欄に reduce y を書き込む．ここで，y は規則 $Y \rightarrow \alpha$ を表す規則番号である．

(4) (3) で生成した LR 表から，制約伝播アルゴリズム [Li 96] を使用して，与えられた T_1, T_2 に属する記号間の接続制約を満たさない動作を除去し，LR 表を可能な限り圧縮する．

以上のアルゴリズムにより生成された LR 表には，GOTO 部が存在しないことに注意．

2.3 LR 表生成の例

ここでは，日本語の動詞を生成する表 5 の文法を例に LR 表の生成を行なう．G_1 は文字列を導出する文法 G_{11} [3]と，細品詞列を導出する文法 G_{12} の 2 層からなる．また，表 2 は文字レベルの接続表を，表 3 は細品詞レベルの接続表をそれぞれ表している．

表 1: サンプル文法 G_1

G_{11}
G_{12}
(1) 動詞 ⟶ 動詞語幹 動詞語尾
(2) 動詞語幹 ⟶ 五段動詞語幹
(3) 動詞語幹 ⟶ 一段動詞語幹
(4) 動詞語尾 ⟶ 五段動詞語尾
(5) 動詞語尾 ⟶ 一段動詞語尾
(6) 五段動詞語幹 ⟶ か
(7) 一段動詞語幹 ⟶ か
(8) 五段動詞語尾 ⟶ け
(9) 一段動詞語尾 ⟶ け る

表 2: 文字レベル (T_1) の接続表

	か	け	る	$
か	0	1	0	0
け	0	0	1	1
る	0	0	0	1

表 3: 細品詞レベル (T_2) の接続表

	五段動詞語幹	一段動詞語幹	五段動詞語尾	一段動詞語尾	$
五段動詞語幹	0	0	1	0	0
一段動詞語幹	0	0	0	1	0
五段動詞語尾	0	0	0	0	1
一段動詞語尾	0	0	0	0	1

G_1 から本アルゴリズムで LR 表を生成した結果得られる GOTO グラフを図 2 に示す．ここで，網掛け部は手順 (2) で拡張された部分である．この時点で，文字 (T_1) レベルの制約だけでなく，細品詞 (T_2) レベルの制約が組み込まれ，この例の場合には I_0 の各アイテムから不必要な先読み記号があればそれを削除す

[3]説明の都合上，終端記号を文字としたが，通常は単語や音素を終端記号の単位とする．

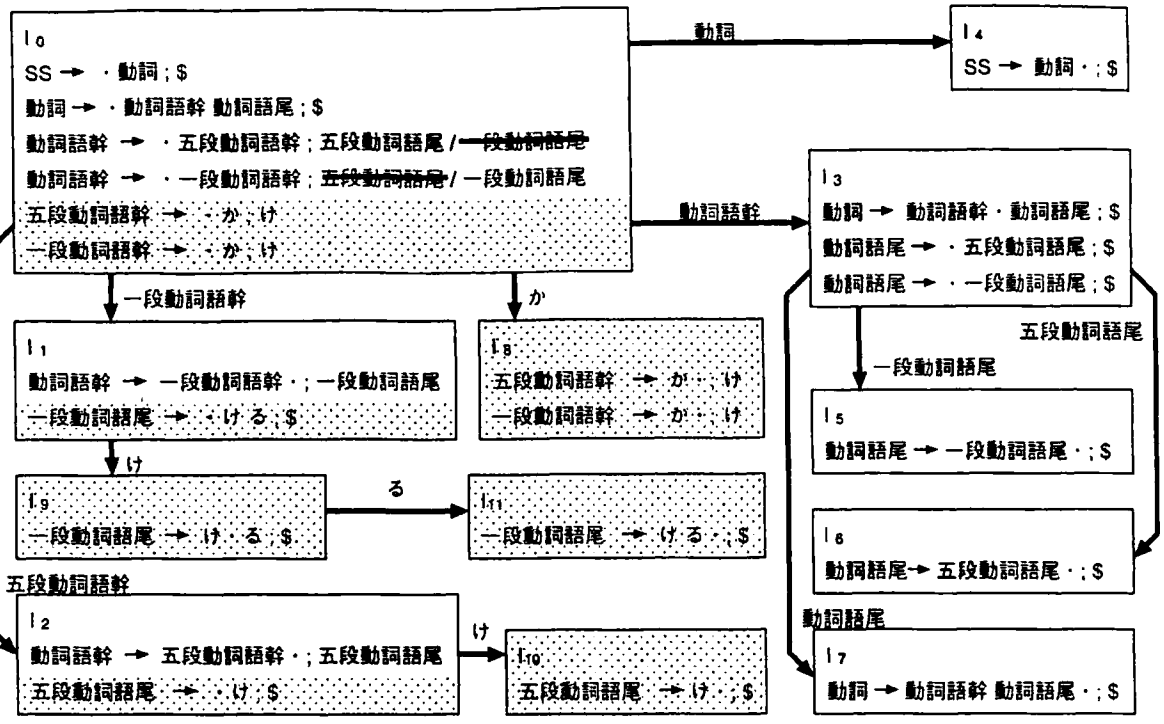

図 2: 複数の接続制約を扱う LR 表生成アルゴリズムで作成した，G_1 に対する GOTO グラフ

表 4: 図 2 の GOTO グラフから生成される G_1 に対する LR 表

	T_1			T_2				N_2			
	か	け	る	五段動詞語幹	一段動詞語幹	五段動詞語尾	一段動詞語尾	動詞語幹	動詞語尾	動詞	$
0	sh8			sh2	sh1			sh3		sh4	
1		sh9					re3				
2		sh10				re2					
3						sh6	sh5		sh7		
4											acc
5											re5
6											re4
7											re1
8		re6/re7									
9			sh11								
10											re8
11											re9

る（線で消した部分）．そして，図 2 から生成される LR 表を表 4 に示す．従来の LR 表に存在した GOTO 部がなくなり，すべて ACTION 部で構成されている．その理由は 3 節で具体的に説明する．

3 複数の制約を含んだLR表に対応するためのGLR法の拡張

図 3 は，G_1 を用いて従来の GLR 法で「かけ」の解析を行なった場合の動作例の概略である（途中省略を含む）．この解析過程には，先読み記号が常に終端記号 T_1（文字）であるため問題が生じる．

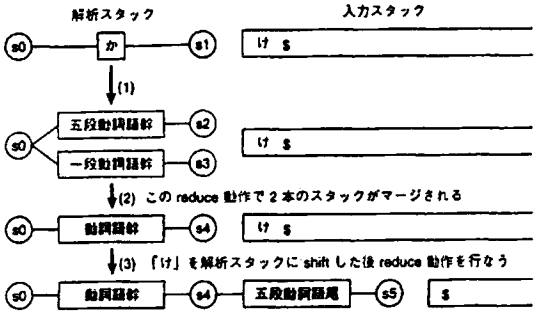

図 3: 従来の GLR 法による解析動作例

過程 (2) において，「動詞語幹」まで部分木が構成された時点でのパーザの状態がいずれも同じ状態になってしまうため，五段動詞語幹，一段動詞語幹のどちらから木が組み上げられたのか区別がつかなくなってしまう．また，その後の解析では，動詞語幹の部分木構成直後の先読み記号は文字の「け」であり，細品詞ではない．よって，五段動詞語尾と一段動詞語尾のいずれの細品詞が後続するのかは，過程 (3) が終了して解析スタックに積まれるまでわからない．その結果，「一段動詞語幹」の後ろに「五段動詞語尾」が接続する，誤った解析が含まれてしまう．

一方，2.2 節のアルゴリズムで生成される LR 表では，非終端記号を先読み記号として shift 動作，reduce 動作を定義することによって，この問題を解決している．それに伴い，解析アルゴリズムにも拡張を施す必要がある．以下に変更点の概要を示す [Walters 80]．

> $X \to \alpha (X \in T_1 \cup N_1)$ なる規則を reduce する時には，スタックをポップした後に通常行なう X をスタックに積む goto 動作を行なわず，代わりに X を入力スタックに押し戻して X を先読みとする動作を引続き実行する．

表 4 を用いて，この拡張アルゴリズムにより「か

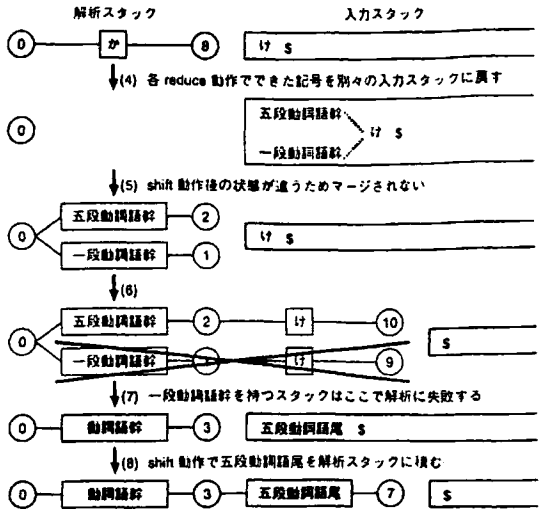

図 4: 拡張 GLR アルゴリズムによる解析動作例

け」を解析した例を図 4 に示す．過程 (4),(5) において，reduce 動作で得られた細品詞（五段動詞語幹，一段動詞語幹）をいったん入力スタックに戻し，五段動詞語幹，一段動詞語幹を先読みとする動作 (shift 2, shift 1) を行なってその後のパーザの状態を区別している．過程 (4) では，2 つの細品詞記号が入力スタックにグラフ状に積まれる．解析スタック（左スタック）だけでなく，入力スタック（右スタック）もグラフ構造化することにより，解析の効率化を図れることに注意されたい．また，過程 (6) では，2 本のスタックがマージされないことにより，一段動詞語幹を要素に持つスタックのみ解析に失敗し，正しい解析スタックだけを残すことが可能となっている．

このアルゴリズムの動作変更の意味を考える．reduce 動作で生成された記号を直接解析スタックに積まずに入力スタックに戻すということは，1 シンボル分だけ解析を遅延させて判断を保留することに相当する．この遅延処理によって，文法の中間層に当たる細品詞 (T_2) レベルの接続制約を扱うことができると考えられる．

図 5 は，表 4 を用いて G_1 から導出可能な構文木を表している．丸で囲まれた数字は，拡張 GLR アルゴリズムによる各動作実行後のスタックトップの状態番号を表し，点線は解析の流れを示している．G_1 から導出可能な 4 種類の木のうち，(c) と (d) は細品詞レベルでの接続に矛盾があり生成されない木である．

図 4 の説明でもわかるように，構文木 (c),(d) の解析において，「か」に対する細品詞記号を shift した後の動作が継続できないため，それぞれ解析が失敗して

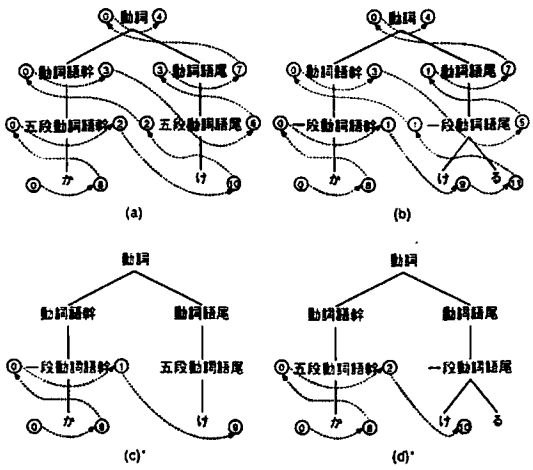

図 5: G_1 から導出可能な構文木

いる．すなわち，図 5 は，2 節，3 節の手法を用いることで細品詞レベルの制約を満たさない構文木の生成を正しく抑制していることを示している．

4 複数の制約を含んだLR表への確率GLRモデルの導入

この節では，PGLR モデル [Inui 97] の定義に若干の変更を施せば，複数の接続制約を含んだ LR 表にも容易に応用可能であることを示す．

4.1 PGLRモデルの定義

ここでは，PGLR モデルの定義の概要のみ示す．PGLR モデルは，LR 表中の各 shift 動作，reduce 動作に対してその生起確率を与える形で定義される確率モデルである．GLR 法では，初期状態から解析成功までに実行された一連の動作により生成される状態遷移系列が 1 つの解析木に相当する．したがって，状態遷移系列の生起確率が構文木の生成確率と等価になる．

パーザのスタックの状態遷移列を T とする．T は式 (4) で表せる．

$$\sigma_0 \stackrel{l_1,a_1}{\Longrightarrow} \sigma_1 \Longrightarrow \cdots \stackrel{l_{n-1},a_{n-1}}{\Longrightarrow} \sigma_{n-1} \stackrel{l_n,a_n}{\Longrightarrow} \sigma_n \qquad (4)$$

ここで，σ_i, l_i, a_i はそれぞれスタックの状態，先読み記号，実行された動作を表している．これを用いて，状態遷移列 T の生成確率 $P(T)$ は

$$P(T) = P(\sigma_0, l_1, a_1, \sigma_1, \ldots, l_n, a_n, \sigma_n) \qquad (5)$$

と表せる．PGLR モデルでは，各解析ステップの実行される確率は，直前のスタックの状態のみに依存するという仮定を導入し，式 (5) を以下のように近似する．

$$P(T) \approx P(\sigma_0) \cdot \prod_{i=1}^{n} P(l_i, a_i, \sigma_i | \sigma_{i-1}) \qquad (6)$$

さらに，LR 表においては，

- l_i と a_i が決まれば σ_i は必ず一意に決まる．
- reduce 動作では先読み記号を消費せず，直前の動作と同じ先読み記号が用いられる．よって，reduce 動作では，先読み記号を予測する必要がない．

という 2 つの特徴があるため，式 (6) の各条件付き確率の推定は式 (7),(8) で行なうことができる．

$$P(l_i, a_i, \sigma_i) \approx P(l_i, a_i | s_{i-1}) \quad (s_{i-1} \in \boldsymbol{S_s}) \qquad (7)$$

$$P(l_i, a_i, \sigma_i) \approx P(a_i | s_{i-1}, l_i) \quad (s_{i-1} \in \boldsymbol{S_r}) \qquad (8)$$

ただし，$\boldsymbol{S_s}$ は shift 動作直後に遷移する状態の集合，$\boldsymbol{S_r}$ は reduce 動作直後に遷移する状態の集合をそれぞれ表す．

式 (7),(8) は，学習データを GLR 法で解析し，解析中に使用された動作を「状態番号, 先読み記号, 実行された動作」の 3 つ組で数え上げることにより，容易に計算できる．

4.2 PGLR モデルの変更

2 節で示した手法で作成される LR 表が従来の LR 表と大きく異なる点は，文法全体 ($\boldsymbol{G_1}$) から見た時には非終端記号となる $\boldsymbol{T_1}$ 以外に属する記号を先読みとする動作が定義されていることである．PGLR モデルは全ての shift 動作，reduce 動作に対して生起確率を与える確率モデルのため，$\boldsymbol{T_1}$ 以外に属する記号を先読みとする動作にも確率を与えなければならない．しかしながら，$\boldsymbol{T_1}$ に属する記号を先読みとする動作と $\boldsymbol{T_1}$ 以外に属する記号を先読みとする動作が，LR 表中の同じ状態に混在する状況は，全ての LR 表に存在する．したがって，確率を与える際の正規化をどのように行なうかが問題となる．

しかし，以下の事実により，$\boldsymbol{T_1}$ 以外に属する記号をを先読みとする動作は全て状態と先読みで正規化すればよいことがわかる．

事実 1: $\boldsymbol{T_1}$ 以外に属する記号を先読みとする動作が定義されている状態は，初期状態を除いて全て $\boldsymbol{S_r}$ に属する．

証明: もし $\boldsymbol{S_s}$ に含まれる状態に $\boldsymbol{T_2}$ を先読みとする動作が存在すると仮定すると，$\boldsymbol{T_2}$ は $\boldsymbol{T_1}$ に属していることになるが，これは式 (1) と $\boldsymbol{G_1}$ の定義から導かれ

る $T_1 \cap T_2 = \emptyset$ に反する．また，N_1,N_2 を先読みとする動作についても同様に背理法で証明できる． □

事実 1 に含まれない例外は初期状態であり，この部分のみ PGLR モデルの定義に変更が必要である．T_1 に属する記号を先読みとする動作は状態のみで正規化し，T_1 以外の記号を先読みとする動作は状態と先読み記号で正規化する．基本的には，この変更のみで複数の制約を含む LR 表に対しても PGLR モデルを適用することができる．

5 音声対話コーパスによる評価実験

5.1 評価方法

ATR の音声対話データベース (SLDB，以下 ATR コーパスと呼ぶ) [Nakamura 96] を使用した実験を行ない，本手法の有効性を評価した．ATR コーパスは 618 旅行対話を収録した約 21,000 文に対して形態素情報と構文木が付与されている．

また，田中らが ATR コーパス用に開発した日本語句構造文法 [田中 97] を使用した．この文法は 441 種の細品詞を終端記号とする 859 規則からなる．我々は，コーパスから獲得した 5,222 単語に対して，各単語を異音列に展開した異音列生成規則をこの文法に追加し，異音レベル，細品詞レベルの 2 層を持つ文法を作成した．文法の緒元を表 5 に示す．また，辞書規則を異音列に展開しない，単語レベル，細品詞レベルの 2 層を持つ文法も同様に作成した（表 6）．これは，音声認識の分野では単語単位の評価が多く行なわれているためである．

表 5: 実験に使用した文法（異音レベル）

規則集合	P_2	$P_1 - P_2$	P_1
規則数	859	6,549	7,408
終端記号（異音）数	—	1,547	1,547
平均規則長	1.39	5.74	5.23

表 6: 実験に使用した文法（単語レベル）

規則集合	P_2	$P_1 - P_2$	P_1
規則数	859	5,222	6.081
終端記号（単語）数	—	4,791	4,791
平均規則長	1.39	1.00	1.05

まず，コーパスから 1 形態素文（「はい」などの相づちが含まれる）を実験データから除去し，残りの文を句構造文法を用いて GLR パーザで解析した．受理できた文のうち，コーパスの形態素・構文情報から半自動的にもしくは人手で正解構文木を付与できた 9,794 文を今回の実験に使用した．この評価データの文の長さ分布を図 6，図 7 にそれぞれ示す．

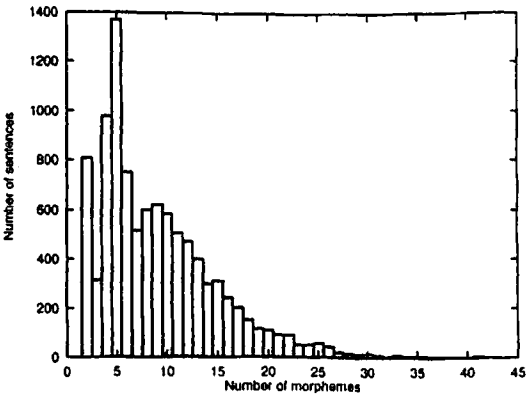

図 6: 評価データの文の長さ分布（形態素数）

次に，異音レベルの制約のみを組み込んだ LR 表と異音レベル，細品詞レベルの制約を組み込んだ 2 種類の LR 表を作成した．そして，これらの LR 表に対して PGLR モデルによる確率値を付与した．

今回の実験では，音声認識への応用を考慮しているため，評価尺度には，音声認識の分野で比較的よく用いられるテストセットパープレキシティ(TSP)[Jelinek 90] を採用した．TSP は言語モデルの複雑さを表し，その値が小さいほど良い言語モデルと評価される．なお，本実験では，PGLR モデルの 1 文の生成確率の値を上位 10 位までの解析木の生成確率の和で近似した．その理由として，曖昧性の多い文の全ての構文木の確率の総和を求めるのが非常に困難であること，下位の木の生成確率値は 1 位のそれと比較して無視できる程度に小さくなると予想できることが挙げられる．予備実験では，1 位の木と 10 位の木の確率値の比の平均は約 2%であった．

そして，現在の音声認識研究で主流となっている N グラム言語モデルと比較するため，バイグラムモデルとトライグラムモデルを異音レベル，単語レベルのそれぞれについて作成し TSP 値を計測した．これらのモデルでは，データ不足を補うために Katz のバックオフスムージング [Katz 87] による平滑化を行なった．

また，評価データの規模が小さいため，データを 10 セットにランダムに分割し，クロスバリデーションにより評価した．データ不足を補うため，PGLR モデルにおいてフロアリングによる平滑化を行ない，フロア値を変化させて，もっとも良い結果を採用した．LR 表の各動作へのフロア値の与え方として，以下の 3 種類

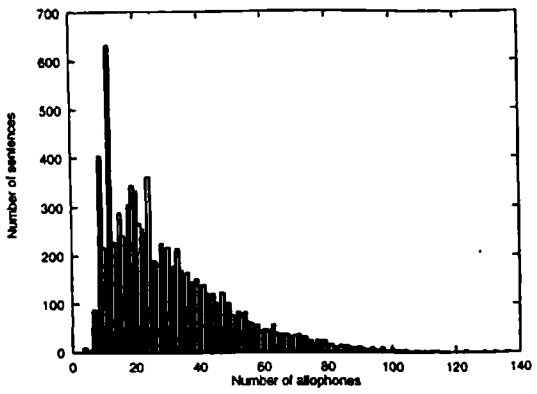

図 7: 評価データの文の長さ分布（異音数）

を試みた．

(1) LR 表中の全ての動作に一律に一定のフロア値を与える．

(2) 競合を起こしている部分のみ 0 頻度の動作を削除し，残りの全ての動作に一律に一定のフロア値を与える．

(3) 0 頻度の動作は全て削除し，フロアリングは行なわない．すなわち，平滑化を行なわない．

なお，単語を文の単位とした出力構文木の正解率を評価尺度とした実験が [今井 99] により行なわれているので，そちらも参照されたい．

5.2 結果と考察

表 7，表 8 に各言語モデルの異音 TSP 値，単語 TSP 値をそれぞれ示す．1-con, 2-con は LR 表に組み込まれた制約の数を表し，括弧内の数字はフロアリング手法を表している．また，TSP 値に付記されている括弧内の値は対応するフロア値を示している．異音レベル，単語レベルの実験のいずれにおいても，組み込んだ制約の数，フロアリング手法によらず，PGLR モデルはトライグラムモデルより低い TSP 値となっていることがわかる．ただし，手法 (3) は被覆率も大幅に低下しているため，手法 (3) の値はあくまで参考程度にとどめておく方がよいだろう．また，PGLR モデル内の各パラメータ間の比較では，複数の制約を組み込むことにより TSP の値が下がっていることが確認できるが，フロアリング手法の違いによる TSP 値の変化はほとんど認められない．

しかしながら，表 9，表 10 から，フロアリング手法の違い，すなわち LR 表から動作を削除する手法の違いにより，平均解析木数に影響を与えていることがわかる．したがって，異音レベル，単語レベルのいずれにおいても，TSP 値が最も低く，かつ被覆率を低下させずに解析木数を削減する 2-con PGLR(2) のモデルが言語モデルとして最も良い性能を示していると考えられる．

表 7: 各言語モデルの異音パープレキシティ

言語モデル	TSP		被覆率 (%)
バイグラム	4.52		100.0
トライグラム	3.26		100.0
1-con PGLR(1)	2.75	(0.1)	99.5
2-con PGLR(1)	2.65	(0.1)	97.1
1-con PGLR(2)	2.75	(0.1)	98.9
2-con PGLR(2)	2.66	(0.1)	97.6
1-con PGLR(3)	2.20		66.6
2-con PGLR(3)	2.10		62.7

表 8: 各言語モデルの単語パープレキシティ

言語モデル	TSP		被覆率 (%)
バイグラム	21.55		100.0
トライグラム	15.47		100.0
1-con PGLR(1)	13.38	(0.3)	100.0
2-con PGLR(1)	11.41	(0.7)	100.0
1-con PGLR(2)	12,89	(0.7)	100.0
2-con PGLR(2)	11.25	(0.7)	99.8
1-con PGLR(3)	7.90		60.1
2-con PGLR(3)	7.11		59.6

PGLR モデルがトライグラムより低い TSP 値を示す要因として，PGLR モデルでは文脈依存性が考慮されていることが挙げられる [Inui 97]．PGLR モデルの構築においては，GLR パーザの状態が考慮される．パーザの状態はその時点までに解析されたスタックの状態に依存するため，左文脈にある程度依存していると考えることができる．本実験結果は，PGLR モデルの文脈依存性がトライグラムモデルにおける直前の 2 シンボル分の文脈情報より強いことを示唆している．

以上の結果より，PGLR モデルは音声認識用の言語モデルとして有効であると期待できる．しかしながら，

表 9: 各手法ごとの最大および平均構文解析木数（異音レベル）

手法	最大	平均
1-con + (1)	2.12×10^{9}†	5.43×10^{7}†
2-con + (1)	6.10×10^{6}	718
1-con + (2)	1.92×10^{9}‡	2.47×10^{6}‡
2-con + (2)	1.58×10^{4}	6.52
1-con + (3)	3.60×10^{3}	5.23
2-con + (3)	32	1.62

†: 解析木数が 4 バイト整数型で扱える最大値を超えた 2,097 文を除いて計算．

‡: 同 36 文を除いて計算．

表 10: 各手法ごとの最大および平均構文解析木数（単語レベル）

手法	最大	平均
1-con + (1)	2.06×10^{9}†	8.17×10^{6}†
2-con + (1)	1.11×10^{5}	45.8
1-con + (2)	3.25×10^{6}	1.04×10^{3}
2-con + (2)	9.63×10^{3}	4.23
1-con + (3)	264	2.94
2-con + (3)	24	1.35

†: 解析木数が 4 バイト整数型で扱える最大値を超えた 69 文を除いて計算.

実際の音声認識においては，解析時に複数の単語仮説が同時に存在するため，入力異音（単語）列が一意に決定している状況よりタスクが難しく，本手法が有効に機能するかは明らかでない．今後，音声認識システムに本手法を組み込んで実験を行なう必要がある．

6 おわりに

本論文では，複数の接続制約を扱うための LR 表生成アルゴリズム，および GLR 法の拡張アルゴリズムをそれぞれ提案した．また，乾の PGLR モデルをわずかな変更で複数の接続制約を含む LR 表に導入可能なことを示した．ATR 対話コーパスを用いた評価実験では，複数の接続制約を含む PGLR モデルはテストセットパープレキシティでトライグラムモデルよりも低い値を示すことが確認された．

音声認識システムへ本手法を導入するまでには，まだいくつかの課題が残されている．

- 今回の実験で用いた評価データは約 9,800 文であり，規模が小さい．評価データを増やし，より大きな規模での実験を行なう必要がある．
- 現在の大語彙音声認識の研究では，語彙数が 5,000 語から数万語のオーダである．現時点では約 5,000 語の語彙数で LR 表を作成できているが，1 万語以上の語彙数で LR 表を作成できるかは未確認である．もし作成できたとしても，計算コストが大きすぎれば実用に堪えない．語彙数を増やして LR 表を生成する実験を行なう必要がある．
- LR 表作成と同様，解析についてもスケーラビリティの問題がある．LR 表作成実験と平行して解析実験を進める必要がある．

これらの課題を考慮しつつ，現在本手法を組み込んだ音声認識システムを構築している段階である．音声認識実験による評価については，機を改めて発表したい．

謝辞

実験に用いた ATR 対話コーパス (SLDB) を提供下さりました ATR の竹沢寿幸氏に感謝致します．また，ATR コーパス用の日本語文法を提供下さりましたランゲージウェアの衛藤純司氏に感謝致します．

参考文献

[Aho 86] Aho, A.V., Sethi, R., and Ullman, J.D. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, 1986.

[Inui 97] Inui, K., Sornlertlamvanich, V., Tokunaga, T., and Tanaka, H. A new formalization of probabilistic GLR parsing. In *Proceedings of International Workshop of Parsing Technologies*, pp. 123–134, 1997.

[Jelinek 90] Jelinek, F. Self-organized language modeling for speech recognition. In Waibel, A. and Lee, K.F., editors, *Readings in Speech Recognition*, pp. 450–506. Morgan Kaufmann, 1990.

[Katz 87] Katz, S.M. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 35, No. 3, pp. 400–401, 1987.

[Kita 91] Kita, K., Kawabata, T., and Saito, H. GLR parsing in hidden Markov model. In *[Tomita 91]*, pp. 153–164. Kluwer Academic Publishers, 1991.

[Knuth 65] Knuth, D.E. On the transition of languages from left to right. *Information and Control*, Vol. 9, pp. 607–639, 1965.

[Li 96] Li, H. Integrating connection constraints into a GLR parser and its applications in a continuous speech recognition system. Technical Report TR96-0003, Dept. of Computer Science, Tokyo Institute of Technology, 1996.

[Nakamura 96] Nakamura, A., Matsunaga, S., Shimizu, T., Tonomura, M., and Sagisaka, Y. Japanese speech databases for robust speech recognition. In *Proceedings of International Conference on Spoken Language Processing*, Vol. 4, pp. 2199–2202, 1996.

[Tanaka 95] Tanaka, H., Tokunaga, T., and Aizawa, M. Integration of morphological and syntactic analysis based on LR parsing. *Journal of Natural Language Processing*, Vol. 2, No. 2, pp. 59–74, 1995.

[Tomita 86] Tomita, M. *Efficient Parsing for Natural Language*. Kluwer Academic Publishers, 1986.

[Tomita 91] Tomita, M. (ed). *Generalized LR Parsing*. Kluwer Academic Publishers, 1991.

[Walters 80] Walters, D.A. Deterministic context-sensitive languages: Part I. *Information and Control*, Vol. 17, pp. 14–40, 1980.

[綾部 98] 綾部寿樹. 複数の接続表の制約の LR 表への組み込みと実装化. 修士論文, 東京工業大学大学院情報理工学研究科, 1998.

[今井 99] 今井宏樹, 白井清昭, 田中穂積. 複数の接続制約を扱う PGLR 法について. 情報処理学会研究報告,NL-130-8, 1999.

[田中 97] 田中穂積, 竹澤寿幸, 衛藤純司. MSLR 法を考慮した音声認識用日本語文法 - LR 表工学 (3) -. 情報処理学会研究報告,SLP-15-25, pp. 145–150, 1997.